FAQ: AWIPS' implementation of secure shell

1. What is the Secure Shell package?

The Secure Shell package (SSH) is a collection of software that provides shell services similar to the rsh/remsh, rcp and rlogin commands. The rsh/remsh, rcp and rlogin commands are considered a less secure method of communicating between computers because these commands do not encrypt their network traffic (including passwords). SSH encrypts the network traffic using Public Key Cryptography 2.

2. What are the SSH equivalent commands to the less secure rsh/remsh, rlogin, rcp, ftp and telnet commands?

The SSH equivalent commands are:

ssh	replaces rsh, remsh, rlogin and telnet
scp	replaces rcp
sftp	replaces ftp

For manual logins or file transfers with passwords, the SSH commands work like their r-command or ftp analogs, and can usually use the same syntax. For logins, ssh functionally replaces telnet although the syntax is different.

3. When using SSH, is the DISPLAY variable required for exporting graphical GUIs from a remote system?

The AWIPS installation of SSH has X tunneling enabled. That means that if you use ssh to log into another AWIPS machine and start an X client, the X protocol will “tunnel” within the SSH protocol and the GUI will start on your local machine. Do not set the DISPLAY variable.

4. What documentation is available for SSH?

Man pages are available for the various commands (ssh, scp, sftp, ssh-keygen). Most UNIX and RedHat Linux books will contain information about SSH. For more in-depth discussions on SSH the book ***SSH, The Secure Shell***, by Barrett and Silverman is recommended.

5. Which implementation of SSH does AWIPS use?

One confusing thing about SSH is that there are two different implementations (SSH and OpenSSH) and two different protocols (protocol 1 and protocol 2) so that when you consult Barrett and Silverman you will, in some cases, find four answers to the same question depending on the implementation and protocol you are using. The AWIPS installation of SSH is OpenSSH protocol 2. At present (OB2) protocol 1 is partially enabled in the AWIPS installation, but it is deprecated and will probably be disabled in a future release.

The AWIPS implementation of SSH is located in the /usr/local/openssh directory. The ssh, scp, sftp and ssh-keygen executables are located in the /usr/local/openssh/bin directory. There are symbolic links for the ssh, scp and sftp commands in the /usr/local/bin directory so that these commands will be available for easy command line use. A symbolic link from /usr/local/bin to /usr/local/openssh was not set up for the ssh-keygen command because the ssh-keygen command should rarely be executed.

6. Does the AWIPS Gauntlet firewall need to be configured to pass the SSH traffic?

Yes, the AWIPS Gauntlet firewall needs to be configured to pass the SSH traffic. The instructions for configuring the AWIPS Gauntlet firewall are in AWIPS System Administration Note 16 which can be found at http://www.ops1.nws.noaa.gov/awipsnew/software/SysAdm16-ssh_S.pdf.

7. What is the plan for implementing SSH in AWIPS?

The SSH suite of utilities has been installed on all AWIPS platforms with AWIPS Build OB2. Prior to installing OB3, the AWIPS Gauntlet firewall must be configured to pass SSH traffic (see question 6). During the OB3 installation, SSH encryption/decryption key files will be regenerated for all the accounts used by AWIPS baseline software (root, fxa, awipsusr, textdemo, ldad and www). In OB3 the baseline AWIPS applications will start using SSH. For example the ldad listener process will use the scp utility instead of the rcp utility to copy products from ls1 to ds1. By OB4 all of the baseline AWIPS software should be using SSH.

8. My local applications use a less secure command such as ftp. What is the deadline for converting my programs to utilize the SSH equivalent command?

When AWIPS OB4 is deployed (deployment currently scheduled to start mid-September 2004) the daemons for telnet, ftp, and r-commands will be removed from AWIPS platforms. All local applications that use automated logins or file transfers must by that time have been converted to use SSH commands.

9. In the AWIPS implementation of SSH is either the .rhosts or the .shosts file required?

One possible SSH implementation method is to utilize the .rhosts or the .shosts file however the AWIPS implementation of SSH will not use either of these files. The AWIPS implementation of SSH ignores both the .rhosts file and the .shosts file.

10. If the .rhosts and .shosts files are not utilized to allow password-less SSH traffic, then what method is AWIPS utilizing for automated file transfers and logins?

AWIPS is utilizing public key authentication to allow automated (no password challenge) file transfers and logins. To support SSH's public key authentication, the following configuration steps must be performed:

1. On the client system (the system which the ssh, scp or sftp is started from), generate a pair of personal authentication keys: one private authentication key and one public authentication key. The file name of the private authentication key file is ***id_dsa***. The file name of the public authentication key file is ***id_dsa.pub***. Both the id_dsa file and the id_dsa.pub file are located on the client system in the $HOME/.ssh directory of the user that initiates the SSH command.

2. On the host system (the system which the ssh, scp or sftp is directed to), append the public authentication key generated from step 1 to the host's list of authorized keys. The file name of the host's public authentication keys is ***authorized_keys2***. The authorized_keys2 file is located on the host system in the home directory of the user account which is receiving the SSH request.

3. On the client system (the system which the ssh, scp or sftp is started from), append the host's public host key to the client's list of known hosts. That list is maintained in the $HOME/.ssh/**known_hosts** file. The host's public host key is generated at SSH install time. It is located in the /usr/local/openssh/etc/ssh_host_rsa_key.pub file. After adding the new public host key to the known_hosts file, that key record must be edited to add the name of the host to the record.

The four configuration files and their locations are:

Client system SSH files	**Host system SSH files**
$HOME/.ssh/***id_dsa.pub***	$HOME/.ssh/***authorized_keys2***
$HOME/.ssh/***id_dsa***	
$HOME/.ssh/***known_hosts***	

During the OB3 installation, the 4 SSH configuration files will be set up for the each of the following baseline AWIPS accounts: root, fxa, awipsusr, textdemo, ldad and www.

11. Given the number of AWIPS systems (ds1, ds2, as1, as2, px1, px2...) at a typical AWIPS site, it looks like it will be a challenge to keep all of the SSH configuration files synchronized. How will the SSH configuration files be synchronized across the various AWIPS systems?

If you consider the baseline AWIPS user accounts such as root, fxa, awipsusr, textdemo, ldad and www, AWIPS sets up the home directories for each of these accounts on directories which are local to the AWIPS system. For example the home directory for the fxa user account is /awips/fxa. The /awips/fxa directory is unique to each AWIPS system: the /awips/fxa directory on ds1 is different than the /awips/fxa directory on ds2. The unique /awips/fxa home directories on each system implies that there may be unique /awips/fxa/.ssh directories to support the SSH configuration files. Maintaining separate /awips/fxa/.ssh directories with each /awips/fxa/.ssh directory having its own set of id_dsa.pub, id_dsa, known_hosts and authorized_keys2 configuration files could become a logistics nightmare.

To overcome the logistical difficulty in maintaining separate id_dsa.pub, id_dsa, known_hosts and authorized_keys2 configuration files on each AWIPS system, the OB3 installation will set up symbolic links to an NFS directory which is accessible to each server and workstation. The SSH symbolic links established in OB3 will be:

User account	**Symbolic link in home directory points to**	
root	/.ssh	–> /home/ssh/root
fxa	/awips/fxa/.ssh	–> /home/ssh/fxa
awipsusr	/awips/fxa/awipsusr/.ssh	–> /home/ssh/awipsusr
textdemo	/awips/fxa/textdemo/.ssh	–> /home/ssh/textdemo
ldad	/awips/fxa/ldad/.ssh	–> /home/ssh/ldad
www		/home/ssh/www

By using symbolic links, there will only be one set of id_dsa.pub, id_dsa, known_hosts and authorized_keys2 configuration files to maintain for each user account.

12. How are the 4 SSH configuration files which support automated, no password challenge use of the ssh, scp and sftp commands created/maintained?

Four configuration files are required by SSH to support automated, password-less use of the ssh, scp and sftp commands. All of the configuration files are located in the .ssh subdirectory which is located in the home directory of the user.

1. The id_dsa.pub file is the public key file. The id_dsa.pub file is created by the ssh-keygen utility. It resides in the .ssh subdirectory which is located below the user's home directory. The id_dsa.pub file resides on the client system that initiates the ssh, scp or sftp command, i.e. the system that you are scping from. The id_dsa.pub public key file should be readable by everyone with file permissions set to 644 (or 444).

2. The id_dsa file is the private key file. The id_dsa file is created by the ssh-keygen utility. It resides in the .ssh subdirectory which is located below the user's home directory. The id_dsa file resides on the client system that initiates the ssh, scp or sftp command, i.e. the system that you are scping from. The id_dsa private key file must be readable by only its owner with file permissions set to 600 (or 400).

3. The authorized_keys2 file is maintained by manually adding the public authorization key from the client's id_dsa.pub file to the host's authorized_keys2 file. The authorized_keys2 file resides on the host system that receives the ssh, scp or sftp request, i.e. the system that you are scping to. The authorized_keys2 file must be writable on by its owner with permissions set to 640 (or 600). When you add someone else's public key to your authorized_keys2 file, you are authorizing that user to access your account. The authorized_keys2 file does not have to be group and world readable.

4. The known_hosts file can be automatically updated on the first time that you ssh into the host system. The ssh process running on the client system will ask you if you want to add the public host key to the client's list of known_hosts. The known_hosts file resides on the client system that initiates the ssh, scp or sftp command, i.e. the system that you are scping from. The known_hosts file must be writeable only by its owner with permissions set to 640 (or 600). It does not have to be group or world readable.

13. Do I need to configure the private and public keys for user accounts?

During the OB3 installation, SSH encryption/decryption key files will be regenerated for all the accounts used by AWIPS baseline software (root, fxa, awipsusr, textdemo, ldad and www).

For non-baseline user accounts, the local system administration will have to generate the SSH encryption/decryption key files. The ssh-keygen executable is used to create both the public key and the private key.

To add another SSH question to this FAQ WEB page, email your question to Wayne.Martin@noaa.gov.

Many thanks to Stowell Davison and Jim Williams for providing the technical answers to these FAQs.